

РАЗРАБОТКА ПРОГРАММНОГО АГЕНТА «СИНХРОКАЛЕНДАРЬ» ДЛЯ ПРОСМОТРА РАСПИСАНИЯ ЗАНЯТИЙ

Аннотация. В статье представлен процесс разработки программного агента «СинхроКалендарь», который предоставляет возможность пользователям удобно взаимодействовать с расписанием через телеграм-бота или приложение календаря.

Ключевые слова: расписание, платформа Modeus, программный агент, календарь, Телеграм, бот.

Введение. С развитием современных технологий образовательные учреждения переходят на новые форматы организации учебного процесса. Одним из значительных изменений стал переход от традиционных бумажных расписаний к их электронному формату. Как описано в работе [1], теперь каждый учащийся вуза или школы благодаря электронному расписанию может в любое удобное время получить доступ к своему графику занятий, просто разблокировав мобильный телефон или открыв личный кабинет на компьютере.

Учебные заведения используют специализированные платформы для оперативной и удобной публикации расписания, а также внесения изменений в него. Среди образовательных учреждений стали пользоваться популярностью сервисы электронного расписания, построенные по технологии SaaS (Software as a Service). После анализа источника [2] было выяснено, что в данной модели пользователи зачастую получают доступ к приложениям через веб-браузер, что обеспечивает удобство использования и не требует установки и обновления программного обеспечения на устройствах.

Популярная среди учебных заведений платформа Modeus также использует концепцию SaaS [3], и предоставляет университетам доступ к ПО через Интернет. Modeus обладает разносторонним функционалом, включающий проектирование инновационных образовательных программ, учет успеваемости обучающихся и формирование гибкого расписания. На веб-сайте пользователи могут просматривать график занятий, оценки, а также записываться на элективные курсы и занятия по физической культуре.

Школьники и студенты представляют собой наиболее широкий контингент пользователей Интернета, электронных устройств и различных приложений. Мессенджер Телеграм с каждым годом становится более популярным не только в России, но и во всем мире. Пользователи данного приложения могут осуществлять обмен сообщениями, создавать каналы для распространения информации среди своих подписчиков, а также использовать и создавать Телеграм-ботов.

Телеграм-боты представляют собой программы, интегрированные в платформу Телеграм, которые автоматизируют определенные задачи и предоставляют различные услуги пользователям через интерфейс мессенджера. Из работы [4] видно, что Телеграм-боты становятся новым и удобным способом взаимодействия с программой на ряду мобильными и веб-интерфейсами. В образовательной среде они могут быть использованы для предоставления студентам информации о расписании занятий и уведомлении об изменениях в графике встреч.

Проблема исследования. Для доступа к информации на сайте Modeus требуется ввод студенческой почты и пароля при каждом посещении платформы, что может увеличивать временные затраты на получение необходимых данных. Периодически на платформе проводятся технические работы, возможен значительный наплыв пользователей, а также сайт требует наличие интернет-соединения для функционирования. Эти факторы могут стать препятствием для студентов и преподавателей, желающих оперативно получить актуальную информацию из системы Modeus.

Модель бизнес-процесса получения информации из системы Modeus в нотации BPMN 2.0 (рис. 1) демонстрирует, что пользователю требуется выполнить 6 шагов для получения информации о встрече на сайте Modeus. Результаты анализа времени, затраченного на процесс получения информации представлен в табл. 1. В качестве анализируемых данных выбраны действия пользователя, ответы мобильного устройства и время выполнения каждого действия. Проведенный анализ позволяет сделать вывод о том, что регулярный вход в аккаунт и медленная загрузка веб-ресурса, вместе с возможными техническими проблемами и помехами с интернет-соединением, значительно затрудняют получение информации о встрече.

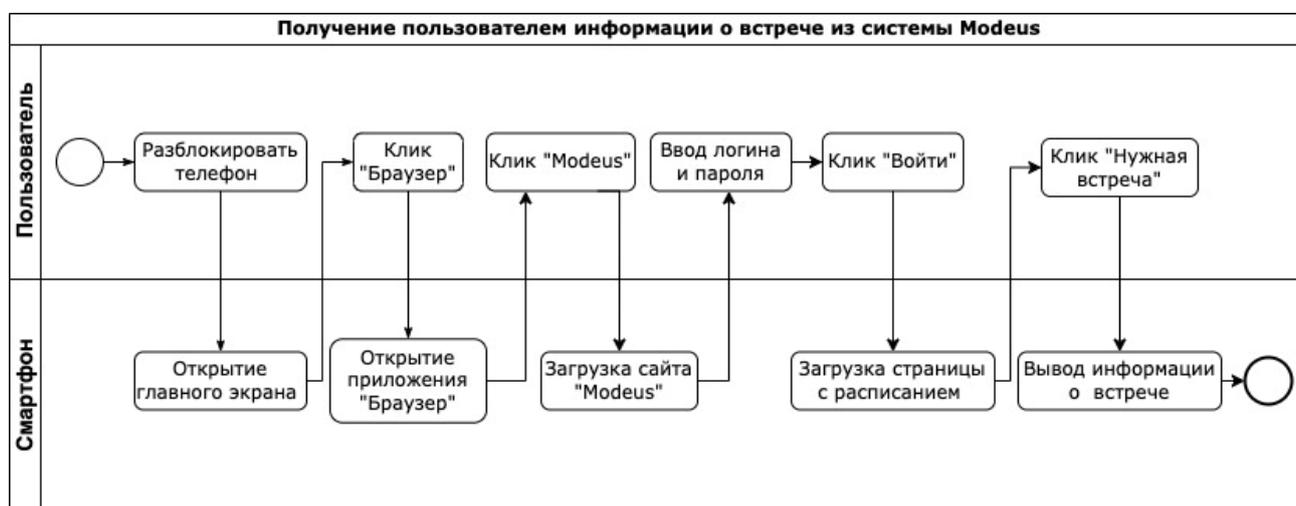


Рис. 1. Модель бизнес-процесса получения пользователем информации о встрече из системы Modeus

Таблица 1

Анализ времени получения информации из системы Modeus

№	Действие студента	Реакция смартфона	Время (секунды)
1	Разблокировать телефон	Открытие главного экрана	3
2	Клик «Браузер»	Открытие приложения «Браузер»	2
3	Клик «Modeus»	Загрузка сайта «Modeus»	5-20
4	Ввод логина и пароля		10-150
5	Клик «Войти»	Загрузка страницы с расписанием	5-30
6	Клик «Нужная встреча»	Вывод информации о встрече	1
Итого			26-206

Важно отметить, что в расписании могут произойти изменения в датах и времени начала занятий. Уведомления об этих изменениях поступают на студенческую почту, однако отсутствуют прямые оповещения о корректировке информации о встречах, что повышает риск пропуска или опоздания студентов или преподавателей.

В университетской среде всегда присутствует определенный процент студентов, активно сочетающих учебу с трудовой деятельностью. Для эффективного управления своим временем они часто прибегают к использованию различных календарей, приложений и заметок. В таких ситуациях пользователи сталкиваются с необходимостью либо ручного внесения учебного расписания в свое приложение, либо постоянного сравнения графика занятий из системы Modeus с собственным графиком. Этот процесс требует значительных временных затрат и, чаще всего, вызывает дискомфорт.

Материалы и методы. В процессе анализа взаимодействия пользователей с платформой Modeus были выявлены проблемы, затрудняющие оперативное получение необходимой информации:

- регулярный вход в личный кабинет, при котором требуется ввод логина (электронной почты) и пароля;
- ограниченная возможность интеграции расписания занятий с личным календарем, что приводит к необходимости регулярного ручного добавления встреч в календарь;
- возникновение технических неполадок и сбоев в работе веб-сайта;
- отсутствие системы уведомлений об изменениях в расписании, за исключением электронных писем;
- необходимость постоянного наличия Интернет-соединения для функционирования сайта.

Разработка программного агента «СинхроКалендарь» позволит устранить перечисленные проблемы. Он представляет собой информационную систему, которая объединяет несколько компонентов, взаимодействующих между собой с целью обеспечения полноценного функционирования приложения.

Разработка программного приложения начиналась с изучения функционала, предоставляемого платформой Modeus, в контексте возможности получения расписания. Одной из основных функциональных возможностей является возможность просмотра расписания других пользователей по их ФИО. Введя ФИО, пользователь может получить расписание, которое имеется у данного человека. Одной из проблем является ограниченная возможность просмотра расписания, доступная лишь по ФИО студента или преподавателя. Для того чтобы определить расписание на основе студенческой почты, требуется осуществить поиск пользователя на платформе Vmeste.

На основе предоставленных данных были разработаны алгоритмы для эффективной реализации и актуализации расписания.

Алгоритм реализации выглядит следующим образом:

Шаг 1. Получение ФИО пользователя с платформы Vmeste, используя студенческую почту.

Шаг 2. Осуществление входа в аккаунт на платформе Modeus.

Шаг 3. Получение расписания, используя ФИО пользователя.

Шаг 4. Производится замена сокращений корпусов на названия институтов, такие как «УЛК-04» в «ФЭИ» и т. д.

Шаг 5. Создание календаря Google и внесение в него расписания при помощи Google API.

Шаг 6. Настройка доступа к календарю, обеспечивая возможность чтения по предоставленной ссылке.

Шаг 7. Генерация уникальной ссылки для подключения.

Алгоритм актуализации реализуется каждый час и представляет собой следующую последовательность:

Шаг 1. Перебор всех зарегистрированных пользователей.

Шаг 2. Получение текущего расписания с платформы Modeus для каждого пользователя.

Шаг 3. Получение существующего расписания из календарей.

Шаг 4. Производится сверка расписаний, выявляя недостающие или лишние занятия.

Шаг 5. Занесение в календарь недостающих занятий и удаление избыточных.

Представленные алгоритмы позволяют эффективно управлять расписанием, обеспечивая актуальность данных для пользователей.

Клиентская часть приложения включает в себя страницу подписки и информационный лэндинг. На рис. 2 в модуле «Первичное подключение» в клиентскую часть изображены следующие информационные потоки:

1) пользователь вводит свою корпоративную почту через Телеграм-бота или web-интерфейс;

2) электронная почта передается на сервер;

3) отправляется запрос с почтой на сервис Vmeste, откуда на сервер приходят ФИО пользователя;

4) ФИО пользователя отправляется на сайт Modeus, откуда на сервер получаем расписание пользователя;

5) сервер с помощью Google API [5] отправляет данные о расписании в Google Календарь, где формируется календарь с графиком занятий и ссылка на него, которая затем отправляется обратно на сервер;

6) сервер передает ссылку на календарь в Телеграм-бот или на web-интерфейс;

7) пользователь получает персональную ссылку на расписание.

Пользователь может взаимодействовать как с web-интерфейсом, так и с Телеграм-ботом. На сервер отправляется запрос на получение данных, и необходимые сведения отправляются пользователю. Используемые технологии в web-интерфейсе включают язык разметки HTML и язык стилей CSS [6] для построения пользовательского интерфейса. Клиентская логика и взаимодействие с пользователем реализованы с применением языка программирования JavaScript [7] и библиотеки React [8].

Серверная часть приложения является неотъемлемой частью проекта. На сервере используется операционная система Ubuntu и веб-сервер Nginx [9]. Серверное приложение построено на основе Node.js [10] и фреймворка Express.js [11] для создания API-приложения, обеспечивающего гибкость и масштабируемость. Взаимодействие с Google-календарем реализуется через Google API (см. рис. 2), что позволяет эффективно обмениваться данными и реализовать чтение/запись событий, создание календарей.

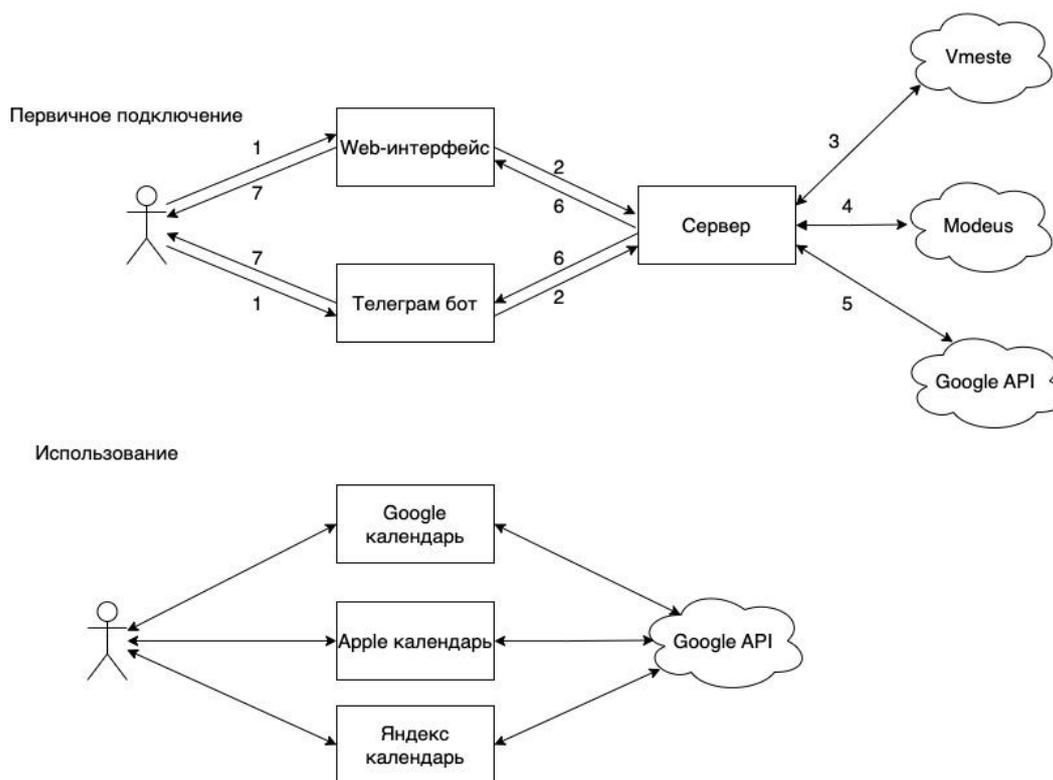


Рис. 2. Архитектура приложения «СинхроКалендарь»

Результаты. При разработке программного агента приоритетом было создание удобного и интуитивно понятного интерфейса для пользователей. Были спроектированы все шаги взаимодействия с программным агентом так, чтобы пользователь мог легко ориентироваться в приложении и выполнять нужные действия.

Основной алгоритм подключения пользователя к приложению «СинхроКалендарь» включает в себя несколько шагов: ввод студенческой почты, получение персональной ссылки на календарь, и подписка на календарь по предоставленной ссылке. На рис. 3 представлена страница формы ввода, куда пользователь должен ввести свою корпоративную почту.

Подключение календаря

Подключить календарь можно на данном сайте или в нашем [телеграм боте](#).
Рекомендуем использовать бота, поскольку в нем больше возможностей.

Для подключения через сайт введите свою корпоративную почту в поле ниже

Корпоративная почта

Рис. 3. Страница ввода электронной почты

После отправки формы выводится окно с персональными данными пользователя и ссылкой на календарь (см. рис. 4). Для пользователей, работающих с Apple Календарем, доступна опция моментального подключения «СинхроКалендаря» через кнопку «Быстрое подключение». При нажатии на нее посредством протокола Webcal автоматически открывается приложение календаря, а ссылка мгновенно передается в форму подписки на календарь.

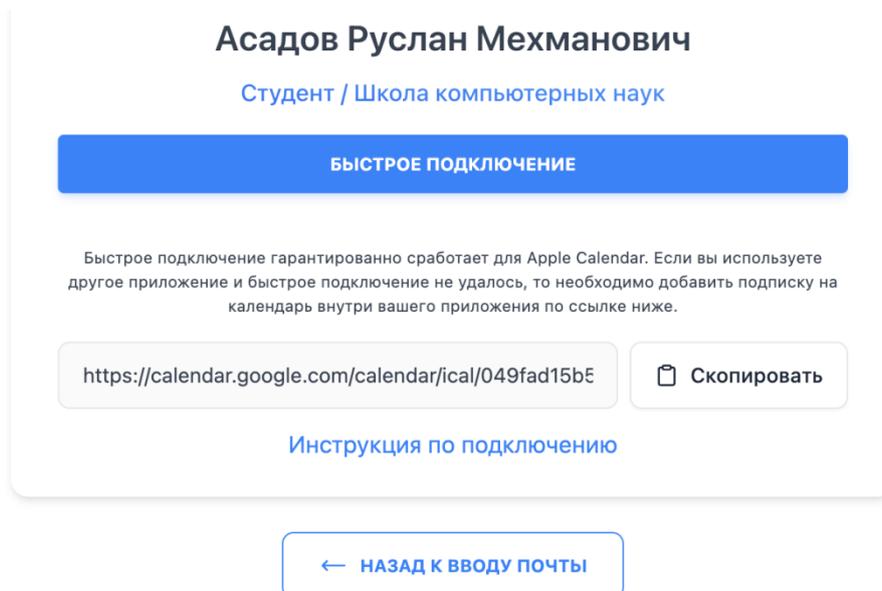


Рис. 4. Форма вывода персональной ссылки

Был разработан Телеграм-бот @modeus_calendar_bot (рис. 5-8), с помощью которого пользователи могут получить ссылку на календарь, просматривать расписание занятий и получать уведомления об изменениях в графике занятий. Взаимодействие пользователя с ботом осуществляется через панель управления внизу экрана.



Рис. 5. Приветственное сообщение телеграм-бота «СинхроКалендаря»



Рис. 6. Сообщение после подключения телеграм-бота

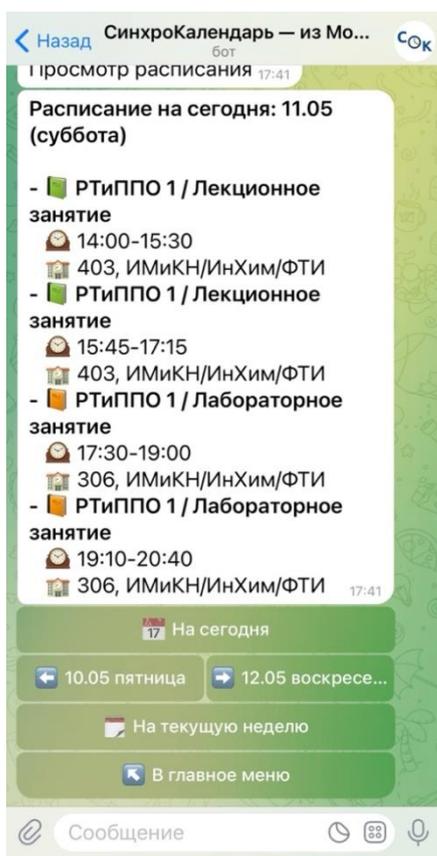


Рис. 7. Просмотр расписания с помощью телеграм-бота



Рис. 8. Полученное уведомление об изменениях в расписании в телеграм-боте

Заключение. В процессе работы над проектом был разработан программный агент «СинхроКалендарь», позволяющий пользователям интегрировать свое расписание занятий из системы Modeus в любой предпочитаемый ими календарь с возможностью подписки по URL. Для большей функциональности был разработан Телеграм-бот, в котором у пользователей есть возможность просматривать расписание по дням и неделям, а также получать уведомления об изменениях в графике занятий.

В будущем планируется масштабирование системы. Таким образом, программный агент будет отвечать не только за работу с расписанием пользователей, но и за координацию других событий и мероприятий университета. Предполагается разработка платформы для отправки приглашений на университетские мероприятия, с помощью которой сотрудники образовательного учреждения смогут приглашать студентов на различные мероприятия. Студенты будут получать уведомления в Телеграм-боте, и в случае заинтересованности смогут автоматически добавить соответствующее мероприятие в свой календарь вместе с расписанием. Это позволит повысить информированность студентов о мероприятиях, проводимых университетом, и потенциально увеличить их заинтересованность в активной учебной и культурной жизни университета.

Кроме того, в Телеграм-бот будут добавлены новые возможности, такие как исключение определенных пар из графика занятий и настройка напоминаний о встречах.

СПИСОК ЛИТЕРАТУРЫ

1. Дрягин П.Е., Иванов А.Г., Кондратенко И.Б. Электронное расписание как необходимая часть современного образовательного пространства вуза // Международный научный журнал «Инновационная наука». — 2017. — № 12. — С. 41-42.
2. Рассудов А.Н., Вайтекунене Е.Л. Применение модели SaaS в России // Актуальные проблемы авиации и космонавтики. Информационные технологии. — 2010. — С. 432-433.
3. MODEUS. Платформа управления индивидуальными образовательными траекториями в университетах и ДПО. — URL: <https://modeus.custis.ru> (дата обращения: 07.04.2024).
4. Козлов А.А., Батищев А.В. Телеграм бот как простой и удобный способ получения информации / Территория науки. — 2017. — № 5. — С. 55-64.
5. Документация Google Calendar API. — URL: <https://expressjs.com/ru> (дата обращения: 09.12.2023).
6. Документация Tailwind CSS. — URL: <https://tailwindcss.com> (дата обращения: 08.05.2024).
7. Современный учебник JavaScript. — URL: <https://learn.javascript.ru/> (дата обращения: 08.05.2024).
8. Документация React. — URL: <https://react.dev> (дата обращения: 08.05.2024).
9. Документация Nginx. — URL: <https://nginx.org/ru> (дата обращения: 08.05.2024).
10. Документация Node.js. — URL: <https://nodejs.org/docs/latest/api> (дата обращения: 08.05.2024).
11. Документация Express.js. — URL: <https://expressjs.com/ru> (дата обращения: 08.05.2024).